Quantum Neural Networks Achieving Quantum Algorithms

Nicolay, Delphine; Carletti, Timoteo

Published in:
Artificial Life and Evolutionary Computation - 12th Italian Workshop, WIVACE 2017, Revised Selected Papers

DOI:
10.1007/978-3-319-78658-2_1

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

Quantum Neural Networks Achieving Quantum Algorithms

Delphine Nicolay(✉) and Timoteo Carletti

Department of Mathematics, Namur Institute for Complex Systems (naXys), University of Namur, Namur, Belgium
delphine.nicolay@unamur.be

Abstract. This paper explores the possibility to construct quantum algorithms by means of neural networks endowed with quantum gates evolved to achieve prescribed goals. First tentatives are performed on the well known Deutsch and Deutsch-Jozsa problems. Results are promising as solutions are detected for different sizes and initializations of the problems using a standard evolutionary learning process. This approach is then used to design quantum operators by combining simple quantum operators belonging to a predefined set.

1 Introduction

Quantum computation has generated a lively interest for the last two decades, since the discovery of a quantum algorithm able to factorize large integers in polynomial time [11]. In fact, the demand for better performance of computers strongly increases and quantum computation could be the answer to overcome the limitations of current computing. However, even in the case of relatively simple problems, the search for a quantum algorithm is not trivial. This fact is clearly illustrated by the parcelled development of solutions for the well known problems of Deutsch [3] and Deutsch-Jozsa [5]. Another complication of quantum computing is its physical feasibility. Indeed, quantum computing requires the development of quantum operators working on systems of qubits. Until now, researchers have been able to physically produce operators dealing with small systems composed of one or two qubits. Fortunately, it has been proved that any quantum operator can be built as a combination of these concretely realizable operators. But, once more, the development of the right combination is not a trivial problem.

In this work, we study the possibility to make use of networks endowed with quantum gates to develop appropriate quantum algorithms, i.e. appropriate combinations of quantum operators to achieve defined tasks or computations. As the construction and the learning process of these networks are roughly inspired by standard artificial neural networks, we decided to name them quantum neural networks (QNN). They are designed for their specific goals by evolutionary optimization methods. The already mentioned Deutsch and Deutsch-Jozsa problems

M. Pelillo et al. (Eds.): WIVACE 2017, CCIS 830, pp. 3–15, 2018.
https://doi.org/10.1007/978-3-319-78658-2_1

have been the first tasks considered for this study. We show that our methodology has led to promising results, as solutions have been detected for different sizes and initializations of the problems. Then, we have identified a set of universal quantum operators and we have applied our method to the design of quantum gates by combining operators from this set. This second phase of the research highlights an important limitation of our model which is the exponential increase of the possible combinations.

The paper is organized as follows. In Sect. 2, we remind the basic concepts of quantum computing and we present our Quantum Neural Network model. In Sect. 3, we detail the problems of Deutsch and Deutsch-Jozsa and results we get with our model. We also perform a critical discussion about our optimization methods. Section 4 presents our attempt of gates development with a set of universal quantum operators. Section 5 concludes the contribution with a summary of our results and perspectives for future work.

2 Background to Quantum Computing

2.1 Quantum Bits

The bit is the fundamental unit of classical computation. Quantum computation is developed upon a similar concept, the quantum bit, also called qubit. These qubits have basic states $|0\rangle$ and $|1\rangle$, which correspond to logical states 0 and 1 for classical bits. But, contrary to the latter ones, qubits can also be in a superposition of states

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where α and β are complex numbers constrained by the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. Usually, a qubit is considered as a vector in $\mathbb{C}^2$ and the basic states are then seen as a pair of orthonormal basis vector

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

As qubits are quantum objects, this superposition of states is not observable. Once the qubit is measured, the superposition is lost and the system will be found in the state $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$.

In the same way, we can define systems with n-qubit as

$$|x_n x_{n-1} \dots x_1\rangle \text{ where } x_i \in \{0, 1\} \text{ for } i = 1, \dots, n.$$

Such states can be written as a tensor product of qubits but quantum computation is much richer. Indeed, thanks to the superposition, a 2-qubit can be in the state

$$\alpha|00\rangle + \beta|11\rangle$$

which can not be constructed using tensor products of qubits. This property of quantum system is called the entanglement [9] and is proper to quantum systems.

2.2 Quantum Gates

Quantum gates, working on a qubit or an n-qubit system, are obtained using unitary operators, hence they are reversible and they respect the normalization condition. They are the basic building blocks, combined to form quantum circuits. Widely used qubit operators and their matrix representation are presented below.

- Identity operator I:

$$\begin{array}{l} I|0\rangle = |0\rangle \\ I|1\rangle = |1\rangle \end{array} \qquad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- NOT operator X:

$$\begin{array}{l} X|0\rangle = |1\rangle \\ X|1\rangle = |0\rangle \end{array} \qquad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- Operator Y:

$$\begin{array}{l} Y|0\rangle = \ i|1\rangle \\ Y|1\rangle = -i|0\rangle \end{array} \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

- Operator Z:

$$\begin{array}{l} Z|0\rangle = \ |0\rangle \\ Z|1\rangle = -|1\rangle \end{array} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- Hadamard transformation H:

$$\begin{array}{l} H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{array} \qquad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- Phase operator S:

$$\begin{array}{l} S|0\rangle = \ |0\rangle \\ S|1\rangle = i|1\rangle \end{array} \qquad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

- $\pi/8$ operator T:

$$\begin{array}{l} T|0\rangle = |0\rangle \\ T|1\rangle = \frac{\sqrt{2}}{2}(1+i)|1\rangle \end{array} \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

The most used 2-qubit operator is the controlled-not operator (C_{not}), also called the 2-qubit XOR gate, which is represented by

$$\begin{array}{l} C_{not}|00\rangle = |00\rangle \\ C_{not}|01\rangle = |01\rangle \\ C_{not}|10\rangle = |11\rangle \\ C_{not}|11\rangle = |10\rangle \end{array} \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Its effect consists in changing the state of the second qubit if and only if the first one is equal to $|1\rangle$. In the same way, we can define other controlled gates by combining this rule and the qubits presented previously. It has been proved [1] that the controlled-not gate combined with all qubit gates form a universal set for quantum computation.

2.3 QNN Model

Our model of quantum neural networks is based on the model proposed by Deutsch [4]. The idea is to build a network whose nodes are quantum gates and connections bring quantum information through qubits. The network is obviously feedforward and the number of nodes is constant in every layer. Quantum neural networks are trained by means of heuristic optimization methods.

3 Deutsch and Deutsch-Jozsa Algorithms

3.1 Problems Description

The Deutsch [3] and the Deutsch-Jozsa [5] problems are basic problems in quantum computing. The Deutsch problem consists in deciding if a binary function $f : \{0, 1\} \to \{0, 1\}$ is constant using only one function evaluation. It is clear that this is not possible in the classical framework, where two function evaluations are needed. To achieve this goal, we have a quantum black box, called oracle, at our disposal. This oracle computes one of the four possible functions, i.e. forming all the possible couples $f(u) = v$ with $u, v \in \{0, 1\}$, by applying an unitary operator U_f defined as

$$U_f(|x\rangle|y\rangle) = |x\rangle|y \oplus f(x)\rangle$$

where $|x\rangle$ and $|y\rangle$ are the qubits of the system. The quantum circuit representing the solution of this problem is presented in Fig. 1. The sequence of operations described in this figure leads to the final state $|\psi\rangle$:

$$|\psi\rangle = \begin{cases} \pm|0\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] & \text{if } f(0) = f(1) \\ \pm|1\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] & \text{if } f(0) \neq f(1) \end{cases}$$

A measure of the first qubit is then sufficient to evaluate if the function is constant ($|0\rangle$) or not ($|1\rangle$).

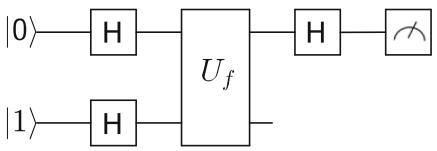

Fig. 1. Quantum circuit for the resolution of the Deutsch problem. The first qubit is initialized to $|0\rangle$ while the second one is set to $|1\rangle$. Then, an Hadamard gate is applied to the two inputs before calling the oracle. An Hadamard gate is finally applied on the first qubit, which is then measured. If it is found in the state $|0\rangle$ then the function is constant, otherwise, namely if the measure determines that the qubit is in the state $|1\rangle$, the function is not constant.

The Deutsch-Jozsa problem is a generalization of the Deutsch problem for a binary function $f : \{0,1\}^n \rightarrow \{0,1\}$. In this case, we have to decide if the function is constant or balanced, which means that we get 0 for half of the function evaluations and 1 for the other half. The resolution is very similar to the previous one and is presented in Fig. 2. Indeed, the qubits are initialized similarly i.e. $|0\rangle$ for the n first qubits and $|1\rangle$ for the last one. Then, an Hadamard gate is applied on all qubits before the oracle intervention. An Hadamard gate operates again on each of the n first qubits. The function is constant if all of them are finally in the state $|0\rangle$.

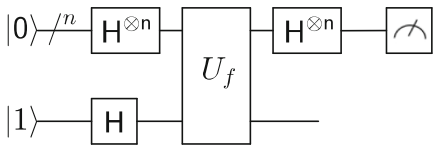

Fig. 2. Quantum circuit for the resolution of the Deutsch-Jozsa problem. The n first qubits are initialized to $|0\rangle$ while the last one is set to $|1\rangle$. Then, an Hadamard gate is applied to all qubits before calling the oracle. An Hadamard gate is finally applied on the n first qubits, which are then measured. If they are all found in the state $|0\rangle$ then the function is constant, otherwise, namely if at least one of the qubits is in the state $|1\rangle$, the function is balanced.

Even if these two problems are relatively simple, let us remark that finding their solution is not trivial. Indeed, the algorithm originally proposed by Deutsch [3] was probabilistic. It was successful with a probability of one half. In [5], Deutsch and Jozsa developed a deterministic algorithm but it required two oracle calls to succeed. The current solution, with only one function evaluation, has been proposed by Cleve et al. [2]. This shows that even in relatively simple cases, there is a need for a general strategy allowing to construct the algorithm associated to the problem at hand.

3.2 Experimentation and Results

For the trial problems of Deutsch and Deutsch-Jozsa, we have not considered a set of universal gates. The nodes could only be assigned to one of the three qubit gates I, X and H or to the oracle. Let us remind that this oracle is only used in one layer of the network, but has an effect on all qubits of the layer. Indeed, our $n+1$ qubits, handled separately, have to be turned into a $(n+1)$-qubit system used as a whole by the oracle. This transformation is carried out using the Kronecker tensor product. The inverse operation is then executed after passing the oracle to recover our $n+1$ qubits.

Quantum neural networks are evolved to solve the considered problem by a genetic algorithm (GA) [6]. The training environment contains the functions

to classify. The fitness of each individual is defined by the fraction of correct classifications. As the optimization is heuristic, all experiments have been replicated 10 times. The results presented are means on these 10 simulations[1].

The first tests on Deutsch problem have been performed with an initialization of the first qubit to $|0\rangle$ and the second one to $|1\rangle$. All simulations led to a correct solution. The only difference observed among these different solutions concerns the operator applied on the second qubit in the last layer, as it is shown in Fig. 3. This difference is not important as only the first qubit is measured to answer the asked question. This solution was already found at the first generation of the GA, this fact could be explained by the small number of possible networks (247).

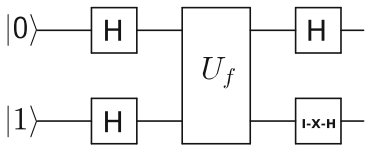

Fig. 3. Quantum circuits for the resolution of the Deutsch problem obtained with our model. The only difference among solutions pertains to the last operator applied on the second qubit, and so has no influence on the state which is measured.

Then, different parameters have been altered to observe the consequences on the learning and the final algorithm. These parameters are the number of layers in the network, the initialization of the qubits and the state to measure to be constant or balanced. When the number of layers is increased, we observe that a solution is always found even if the number of possible networks increase exponentially. Indeed, the number of admissible solutions also increase exponentially according to the number of layers. For example, if we consider five layers in the network, the two networks presented in Fig. 4 have the same effect on the quantum states.

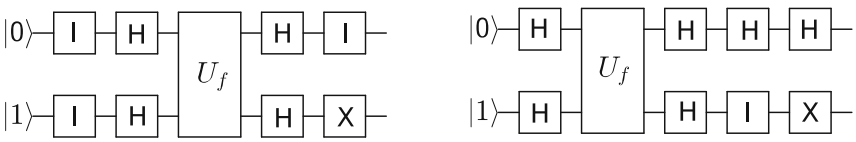

Fig. 4. Two different solutions for the problem of Deutsch if the network is formed by five layers. The networks are different but their effect on quantum bits are equivalent.

[1] The selection is performed by a roulette wheel selection. The genetic operators are the 1-point crossover and the uniform mutation. Their respective rates are 0.9 and 0.01. The population size is 100 and the maximum number of generations is 10000. The survival of best individuals is ensured by elitism.

If we exchange the initialization of the two qubits, we have to consider a network of at least four layers to find a solution. And, most of the time, the solution consists in replacing the network in the previous initialization, which means that a NOT operator is applied to each qubit in the first layer. Results are similar if we alter the initialization by setting both qubits to $|0\rangle$ or $|1\rangle$.

In case we switch the states to measure to have a constant ($|1\rangle$) or balanced ($|0\rangle$) function, we can find a solution whatever we take as initialization of our qubits. The smallest network, given in Fig. 5, is obtained if both qubits are initialized to $|1\rangle$. In other cases, the solution is made of four layers. We have also tried to look for a solution if we measure the second qubit instead of the first one but it has not worked whatever the considered initialization and configuration. This result seems consistent as such a solution has never been introduced in the literature.

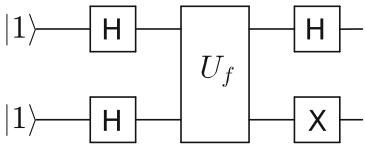

Fig. 5. Quantum circuit for the resolution of the Deutsch problem if a constant function is given by a measure of the first qubit equal to $|1\rangle$. Even if less frequently met, this scheme has already been presented in the literature [8].

Concerning the Deutsch-Jozsa problem, we have tested different sizes of the problem. Let n be the number of variables of the function, then the number of input states of the function is 2^n and the number of possible balanced functions is given by the number of combinations of 2^{n-1} units taken among 2^n. Figure 6 presents the number of possible networks according to n and the mean number of generations to reach the solution with our GA for each of this dimension. We can see in our two graphs that the increase according to n is exponential.

From $n = 3$, we have remarked that our $(n + 1)$-qubit systems could not always be split into $n + 1$ qubits. This is due to the property of entanglement of quantum states. Indeed, some qubits that are combined with the tensor product are modified by the oracle in such a way that they can no more be separated properly. In this case, we have considered either to keep all functions or to exclude functions that lead to entangled states. In the first case, we could hardly get a fitness of 1. In the second case, we have obtained a fitness of 1 but simulations were longer as a preliminary test was needed to remove this type of functions.

3.3 Discussion on the Used Optimization Methods

Before going further, we have considered the possibility of using optimization methods different from genetic algorithms. In this way, we have implemented a simulated annealing (SA) [7] and a random search (RS). Figure 7 shows the

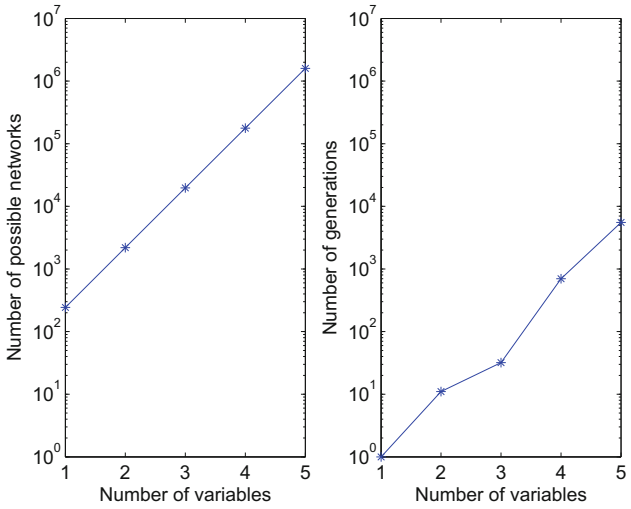

Fig. 6. Number of possible quantum networks (left panel) and number of generations to reach the solution for the Deutsch-Jozsa problem (right panel) according to the number of variables in the function.

number of iterations required by each method to reach the solution for different sizes of the Deutsch-Jozsa problem. We can observe that these numbers are very similar for the random search and the simulated annealing. Regarding our genetic algorithm, the number of required iterations is divided by a factor 100. However, this smaller number of iterations is offset by the number of function evaluations at each iteration, which is 1 for RS and SA and 100 for GA. In conclusion, the genetic algorithm and the simulated annealing do not appear more efficient than the random search.

This fact could be explained by our way of coding and modifying our model of quantum neural networks. Indeed, the shift of the oracle from one layer to another because of the application of a mutation for the GA leads to important changes in networks. This remark also holds for SA, as the oracle can be shifted during the exploration of the space of solutions. Because networks are pretty small, these big changes can modify them as strongly as it is made by random search.

Another explanation could be glimpsed by the analysis of two indicators, namely the fitness distance correlation coefficient and the autocorrelation of the function landscape [7]. As it is indicated by its name, the fitness distance correlation coefficient measures the correlation between the objective function of a candidate and its distance to the optimal solution. As for the autocorrelation, it measures the correlation between neighboring candidates. Results of these two measures for different sizes of the Deutsch-Jozsa problem are presented in Fig. 8. We can see that these two coefficients are quite low, whatever the size of the problem. This observation reinforces our intuition that GA and SA are

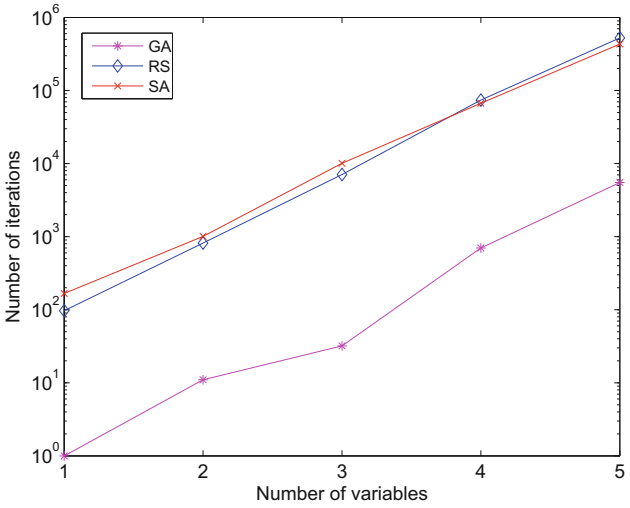

Fig. 7. Comparison of the number of iterations required by each algorithm to reach the solution. This comparison is performed for different sizes of the Deutsch-Jozsa problem. The simulated annealing and the random search required similar number of iterations while it is divided by a factor 100 for the genetic algorithm.

no more efficient than RS for this application. Indeed, if correlation does not exist between the distance to the solution and the objective function, it can not be assumed that the best individual will be found by crossovers and mutations on good individuals. Similarly, the absence of correlation between neighbors removes any advantage to an optimization method such as SA that travels from one candidate to its neighbors.

4 Quantum Gates Construction

Our methodology enables us to develop quantum algorithms solving problems of Deutsch and Deutsch-Jozsa without requiring any particular knowledge except the function to reproduce. Indeed, the appropriate algorithm appears following the learning process applied to a network composed by standard gates. Given the difficulty to develop quantum algorithms and the small number of such algorithms, we think that our results are promising even if the increase according to the number of variables is exponential. Consequently, we have considered to exploit our methodology for the implementation of quantum gates.

Our idea was to identify a set of universal gates and to develop other gates by combining those belonging to this set. We followed the statement of Nielsen and Chuang [10] and worked with a set made of 6 qubit gates to whom the controlled-not gate has been added. The qubit gates are I, H, S, T and their adjoint. As I and H are self-adjoint, we only have to add S^* and T^*.

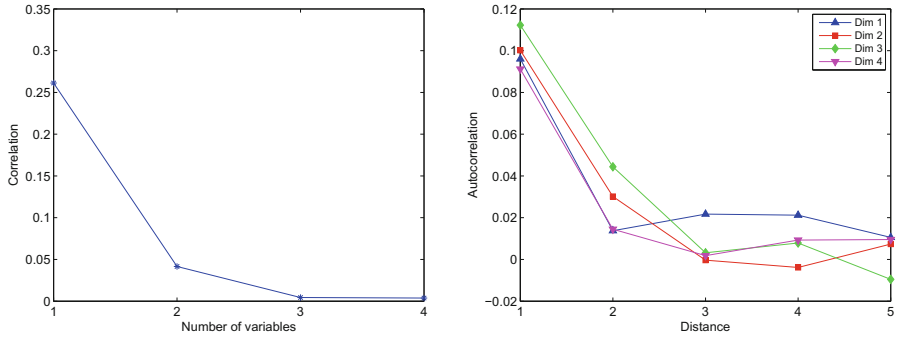

Fig. 8. Indicators analysis for different sizes of Deutsch-Jozsa problem. Left panel: fitness distance correlation coefficient. In our case, the distance between two quantum gates is fixed to 1. Moreover, we do not consider the last operator applied on the last qubit as it has any influence on the final result of the algorithm. Right panel: Autocorrelation of the objective function landscape. For this measure, we consider neighbors at distances from 1 to 5.

Before starting our optimizations, we have analyzed the two indicators presented above in order to choose the most appropriate method. For this, we have considered the objective function of the controlled-Z gate and the Toffoli gate, which is a generalization of the controlled-not for three qubits. The correlation coefficients for these two problems are respectively equal to 0.1048 and 0.2010. The autocorrelation of the function landscape is represented in Fig. 9. Once more, these measures are pretty low. Consequently, we have decided to replace our genetic algorithm by a simulated annealing. Indeed, the genetic algorithm requires more CPU time due to crossover and mutation process for analogous results. Our simulated annealing has a temperature that decreases very slowly[2], with the aim to explore the space of solutions as much as possible.

Firstly, we used our QNN model and our simulated annealing to design the qubit gates that were not part of the defined set, i.e. the X (NOT), Y and Z gate. The Z gate is quite easy to rebuild as it only requires a sequence of two Hadamard gates. On the contrary, X and Y respectively claim 4 and 6 layers and are represented in Fig. 10. Such a number of layers seems quite expensive for so simple gates. Then, we have succeeded in recreating the 2-qubit gates controlled-Y and controlled-Z, which are also represented in Fig. 10. Although it has been proved theoretically that all these gates could be rebuilt from a set of universal gates, let us note that we hereby provide their explicit scheme for the first time.

Nevertheless, we have quickly been confronted to one big limitation of our model, which is the exponential increase of the number of possible networks

[2] The temperature is initialized to 1, in such a way that a candidate decreasing the objective function by 0.5 has a probability of $\frac{2}{3}$ to be accepted. The cooling parameter is fixed to 0.99995 for a slow diminution of this probability.

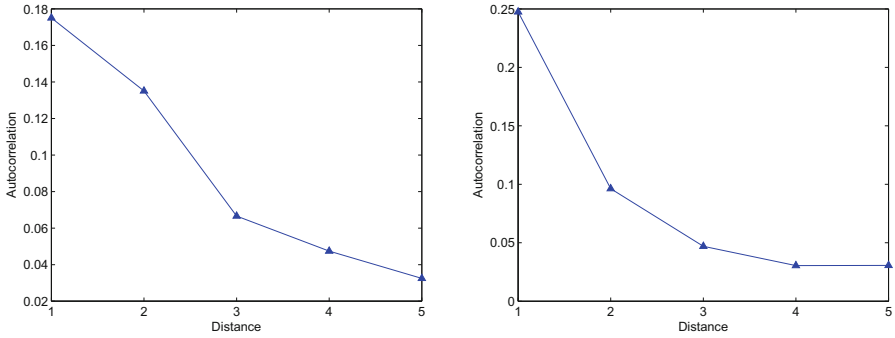

Fig. 9. Autocorrelation analysis for the objective function of two quantum gates. We consider neighbors at distances from 1 to 5. Left panel: Controlled-Z gate. Right panel: Toffoli gate.

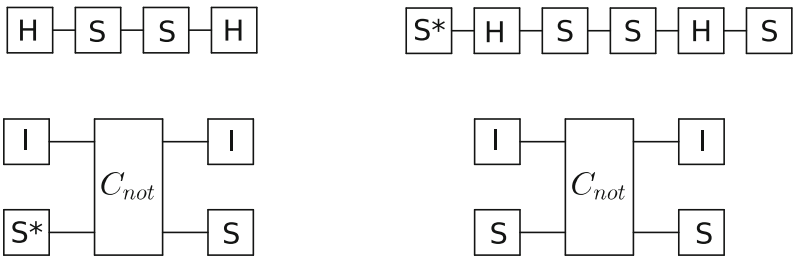

Fig. 10. Design of qubit gates with our model starting from the set of universal quantum gates. Right panel: Not (X) and controlled-Y gates. Left panel: Y and controlled-Z gates.

according to its size. Indeed, we know that a Toffoli gate requires 13 layers of three qubits to be designed from our predefined set [10]. With our model, even if we consider that we know the number of needed controlled-not gate, the number of possible networks among which the solution has to be found is superior to 10^{26}.

5 Conclusion

Quantum computation attracts considerable interest as it can be an answer to the limitations of current computers. Nevertheless, it remains difficult to elaborate quantum algorithms or quantum operators working on systems made of more than two qubits. Our aim is to study the possibility to develop a general framework based on neural networks endowed with quantum gates and evolutionary computation to tackle this difficulty.

Our approach was first used on the Deutsch and Deutsch-Jozsa problems. Results are positive as solutions were found for different configurations and different sizes of these problems. However, we have observed that our optimization method, a genetic algorithm, was no more efficient than a random search among

the space of solutions. This fact can be explained by the low values of the fitness distance correlation coefficient and the autocorrelation of the landscape, as well as by our way of coding the networks. In a second time, our QNN model has been trained to achieve quantum gates from a set of universal quantum gates.

This research highlights two limitations of our approach. The first one is linked to the entanglement property of quantum systems. Indeed, once a state is turned into a entangled state by an oracle or a controlled-not gate, we are no longer able to manage with it. The second one, and the most important for us, is the exponential increase of the networks number according to the size of this network. This increase, combined with the absence of correlation given by our indicators for the objective function, makes the resolution impossible in reasonable time for networks with more than about 15 gates.

Despite these limitations, we can envisage to improve the efficiency of our method. Firstly, we can decrease the number of possible networks by fixing the number of controlled-not gates, and stronger, by fixing the number of one qubit gates that differ from the identity. But, even with these constraints, the size of the resolvable networks will be limited. Another option would be to add a quantum operator to our set as soon as we find its breakdown. Improvements can also be imagine on the learning process. For example, we can consider the addition of a penalty in order to avoid useless sequences of operations.

Acknowledgements. We thank Andrea Roli for his critical reading of a preliminary version of the paper.

This research used computational resources of the "Plateforme Technologique de Calcul Intensif (PTCI)" located at the University of Namur, Belgium, which is supported by the F.R.S.-FNRS.

This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimisation), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

References

1. Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J.A., Weinfurter, H.: Elementary gates for quantum computation. Phys. Rev. A **52**, 3457–3467 (1995). https://doi.org/10.1103/PhysRevA.52.3457
2. Cleve, R., Ekert, A., Macchiavello, C., Mosca, M.: Quantum algorithms revisited. Proc. Roy. Soc. Lond. A: Math. Phys. Eng. Sci. **454**(1969), 339–354 (1998). https://doi.org/10.1098/rspa.1998.0164
3. Deutsch, D.: Quantum theory, the Church-Turing principle and the universal quantum computer. Proc. Roy. Soc. Lond. A: Math. Phys. Eng. Sci. **400**(1818), 97–117 (1985). https://doi.org/10.1098/rspa.1985.0070
4. Deutsch, D.: Quantum computational networks. In: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 425, pp. 73–90. The Royal Society (1989). https://doi.org/10.1098/rspa.1989.0099

5. Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Proc. Roy. Soc. Lond. A: Math. Phys. Eng. Sci. **439**(1907), 553–558 (1992). https://doi.org/10.1098/rspa.1992.0167
6. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Boston (1989)
7. Hoos, H.H., Stützle, T.: Stochastic Local Search: Foundations and Applications. Elsevier, Amsterdam (2004)
8. Mermin, D.: Calculs et algorithmes quantiques: méthodes et exemples. EDP Sciences, Les Ulis (2010)
9. Mintert, F., Viviescas, C., Buchleitner, A.: Basic concepts of entangled states. In: Buchleitner, A., Viviescas, C., Tiersch, M. (eds.) Entanglement and Decoherence. LNP, vol. 768, pp. 61–86. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88169-8_2
10. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
11. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. **41**(2), 303–332 (1999). https://doi.org/10.1137/S0036144598347011